An Evaluation of Techniques for Clustering Search Results[1]

Anton V. Leouski and W. Bruce Croft

Computer Science Department
University of Massachusetts at Amherst
Amherst, MA 01003
leouski@cs.umass.edu, croft@cs.umass.edu

Abstract.

The ability to effectively organize retrieval results becomes more important as the focus of Information Retrieval (IR) shifts towards interactive search processes. Automatic classification techniques are capable of providing the necessary information organization by arranging the retrieved data into groups of documents with common subjects.

In this paper, we compare classification methods from IR and Machine Learning (ML) for clustering search results. Issues such as document representation, classification algorithms, and cluster representation are discussed. We introduce several evaluation techniques and use them in preliminary experiments. These experiments indicate that the proposed techniques have promise, but it is clear that user experiments are required to carry out more thorough evaluation.

---

[1]This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

This material is based on work supported in part by NRaD Contract Number N66001-94-D-6054.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Evaluation of Techniques for Clustering Search Results	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Space and Naval Warfare Systems Center,53560 Hull Street,San Diego,CA,92152-5001**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**19**	

1 Introduction

An IR system typically produces a ranked list of documents in response to a user's query. These documents are presented to the user for examination and evaluation. Although the documents are ranked, there is significant potential benefit in providing additional structure in long retrieved lists. The role of information organization becomes even more important in the interactive model of retrieval, where the focus is on the user's participation in a cycle of query formulation, presentation of search results, and query reformulation.

A natural alternative to ranking is to divide (or cluster) the retrieved set into groups of documents with common subjects. For example, consider a situation when the system is presented with a general query. The retrieval results would contain a wide variety of topics in that general area. An automatic classification tool could create classes of similar documents allowing the user to focus on a particular topic. In this paper we consider the problem of design and evaluation of such a browsing tool for an existing IR system.

We begin by discussing the recent research on clustering in IR and ML. Surprisingly, only a few systems have used clustering methods for organizing retrieval results. Moreover, there is virtually no literature about attempts to evaluate these techniques. Clustering has also been studied in Machine Learning (ML) for a relatively long time and a large number of algorithms has been developed. There has, however, been few application of these techniques to IR [1].

We believe there are four major issues need to be considered:

- the **input** of the classifier, or the document representations. In general, documents are treated as vectors of weight-term pairs. However, the questions of which terms to chose and whether to use the whole document or only a part of it as the source of terms remain to be investigated.
- the classification **algorithm**. The existing clustering techniques vary in accuracy, robustness, speed and storage requirements, etc. More evaluation is needed to choose the appropriate classification procedure for this task.
- the **output** of the classifier, or cluster representations. The classification process results in a set of clusters, where every cluster contains documents about a unique topic. Clusters have been represented using a selected document or term list, but it is not clear that these are satisfactory.
- the **evaluation**. After the classification tool is created, we need techniques to analyze and evaluate its performance from the point of effectiveness and efficiency. The evaluation of the effectivness of an interface tool is much more difficult than the typical retrieval scenario.

The first three of these issues are covered in the third section. We select and present several different classification algorithms and methods for constructing the document vectors and cluster descriptions.

Section 4 deals with the fourth item. It points out the difficulties that arise during evaluation of the clustering techniques, defines the experimental domain, and presents two different approaches for quantitative evaluation of the system performance.

Section 5 describes the experiments we conducted to compare the different methods. It presents and analyzes the results of the experiments. Section 6 summarizes the major results and discusses shortcomings.

One important question, which is not a focus of this paper, is the user interface aspect of the design. Fig. 1 shows a prototype of the interface that we created for our experiments. The titles of the retrieved documents are organized in the tree-like structure that mimics the classification hierarchy. In Section 7, we introduce some of the ideas we are planning to implement in the future.

2 Clustering Techniques

2.1 Document Clustering in Information Retrieval

The basis for using document clustering in IR is the Cluster Hypothesis of van Rijsbergen [2]: *closely associated documents tend to be relevant to the same request*. The idea is that the relevant documents are more similar to each other than to nonrelevant documents. If this hypothesis holds on a particular collection then it would make the retrieval more effective, because the class once found will contain only the relevant documents.

Another way of using document clustering is to give the user the ability to browse through the classification structure, exploring different areas in the collection. This is very helpful in a situation when a user has an information need that he has trouble expressing. Moreover, the user may not be looking for anything specific at all, but rather may wish to explore the general database contents.

Two approaches have been used for document representation. First, the similarity of several documents is measured by a number of citations they have in common. The second, more common way is to represent the documents by a set of manually or automatically assigned index terms.

The basis of every classification process is an association measure among objects. Usually this is a binary relationship that characterizes the dissimilarity between a pair of documents. One may also consider the dissimilarity function as a "distance" function in document hyperspace [3].

Regardless of what similarity measure is used, two basic clustering methods have been carried out to group similar documents: nonhierarchic and hierarchic. Nonhierarchic methods divide a collection into a series of subsets. The most common approach tries to partition N objects into K classes in a way it would minimize the distance of objects to the K centroids [15, 21]. These methods are attractive because of their low computational cost of order $O(N)$ to $O(N \log N)$. However, the resulting structure completely depends upon the choice of the K centroid objects and often fails to reflect the underlying structure of the dataset.

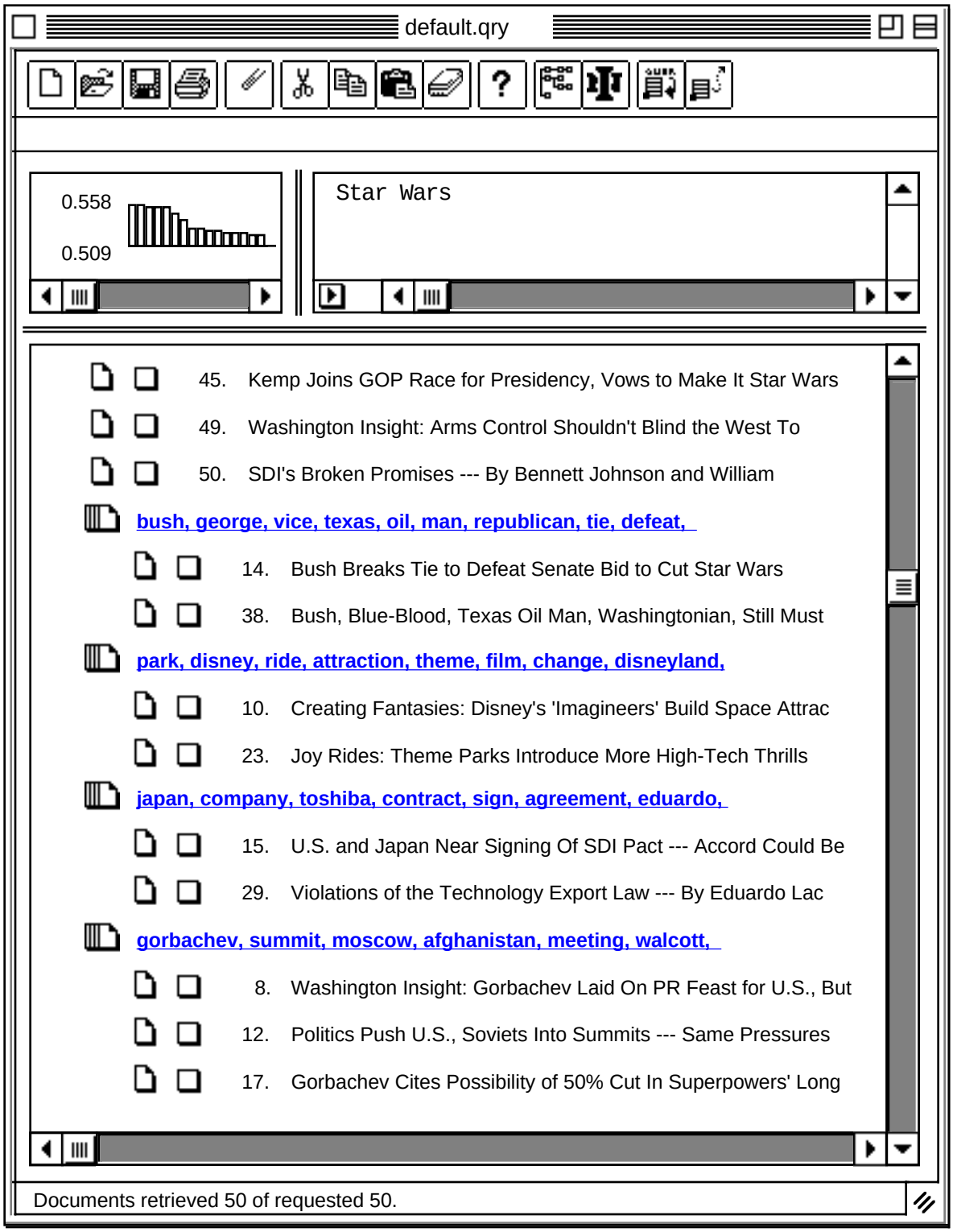

Fig. 1. Shows an example of InCLASS browser window.

Hierarchic methods result in a treelike construction where clusters of closely related documents are nested within bigger clusters containing documents that are less similar. There are two strategies

available for hierarchical clustering. A divisive strategy proceeds by subdividing the initial cluster into smaller and smaller groups of documents. An agglomerative strategy proceeds by building the classification tree bottom-up, joining single documents into the clusters with the whole collection as the tree root at the very end.

The Scatter/Gather system [4] is an example of the browsing approach to the retrieval process. It uses fast document clustering to produce table-of-contents-like outlines of large document collections. During a session with the system, a user is presented with a set of clusters. He chooses several of them as potentially interesting. The documents in these classes are pulled together and reclustered on the fly, to produce a set of clusters covering the reduced collection. In [5], this technique is also used to cluster retrieval results.

2.2 Clustering in Machine Learning

Machine Learning is concerned with improving inference by automating knowledge acquisition and refinement. This process can be described as generalization from particular cases. It begins with objects and creates one or more *classes* or *categories*. An inference element then uses such categories to make inferences about new examples based on partial information. In the domain where events are not pre-classified, an automatic inductive system *learns from observation* (as oppose to *learning from examples*) by recognizing regularities among objects and organizing them into a hierarchy of classes. This learning task was named *conceptual clustering* by Michalski [6].

In cluster analysis, the similarity measure between two objects maps 'distance' between symbolic object descriptions to a single number. The mapping is context-free; the similarity between two objects depends solely on properties of these two objects. This measure is not influenced by the 'environment' surrounding the objects (however, see [7]) or any of external concepts that might be useful to interpret the object configurations. The conceptual clustering systems are attempting to recognize certain configuration concepts. For every cluster such system creates a conceptual description that characterizes the cluster content. Conceptual clustering methods do not form clusters unless they possess concepts that provide some meaningful data interpretation.

Conceptual clustering has both positive and negative characteristics from an IR perspective [8, 11]. One of the goals of this paper is to evaluate a conceptual clustering algorithm for the task of clustering retrieved objects.

3 The Experiments

3.1 Document Representations

For the purpose of clustering we defined a document as a set of term-weight pairs, i.e., as a sparse, high-dimensional vector or profile. In the same way, a group of such objects might be described by combining the profiles of its members.

For both efficiency and effectiveness reasons, the number of terms used to describe documents can be reduced. First, we decided to limit the document representations to only the most essential part of the text. Suppose every document has some kind of abstract, a short summary, or an extended keyword list, that either clearly marked in the text or might be easily pulled out from the rest of the document. The very purpose of such an abstract is a compact definition of the document content and it would serve as a perfect document representative. For example, the first paragraph from a typical magazine article usually contains a rather detailed summary of its content. During our preliminary experiments we observed very reasonable classes by clustering documents from Wall Street Journal articles in 1987 (WallStreet87 database[2]) using only the first two paragraphs. Moreover, in this case the time required to parse the text and build the profiles is fixed and independent from the actual document size. Unfortunately, this approach depends upon the availability of the abstract or some other knowledge specific to the particular database domain. The question was how to find the most essential part of the document without this information? We believed that the query might guide us in this search. During the retrieval process, the query matching algorithm can be applied only to portions of document text and results in a ranked list of the best passages [9]. Fixing the size and the number of passages to represent the document could focus our attention on the most important parts of the text and reduce size differences.

Our experiments showed that this approach introduces a significant amount of confusion into the classification structure. Indeed, we required the system to bring similar documents together. The documents ended up clustered by the content of their best passages, which is usually different from the content of the whole document. It is to no surprise that evaluation methods based on the similarity among documents reported failure.

An alternative is to create complete document vectors and then truncate them. The underlying idea is to select the most important terms and discard unimportant ones. Very rare terms could be removed from the profiles, because they generally do not carry much of the meaning of the document. On the other hand, terms that are very frequent in the retrieved set can also be discarded. The following procedure describes how we constructed document profiles.

---

[2]The WallStreet87 database is a subset of the TREC database [24].

For every document vector we begin by removing every term that appears in more than fifty percent of the documents in the retrieved set. This step is similar to stopword elimination, performed by most IR systems. The rest of the terms are sorted, and a fixed number of words (N_{use}) is selected from the top of the list. The remaining part of the vector is discarded. Finally, from the created profiles we remove several of the most frequent terms in retrieved set ($N_{discard}$).

Several different weighting schemes were tested to rank the terms. We began with the simple term frequency factor. Next, we considered $tf * idf$, $idf = \log\left(N/df\right)$, where tf is the term frequency, df is the document frequency, and N is the number of documents in the test set. Finally, we experimented with the discrimination value (dv) method [3]. The last scheme defines the term weight as $tf * dv$, where dv for a particular term characterizes how much the documents are going to be pull apart if we add this term to the document profiles.

Finally, most documents have titles assigned to them. A title seems to carry additional contextual importance compared to the rest of the document. This property of the title terms could be represented by increasing their weight values in the profile. The extreme case would be to discard the text body altogether and cluster the documents using only title terms.

3.2 Algorithms

All algorithms that we present in this paper are agglomerative and hierarchical. They have time complexity $O\left(n^2\right)$, where n is the number of retrieved documents. Our choice was motivated by the idea that these algorithms are generally more robust and independent from the order in which the objects are presented. Although the processing time rises quadratically with increasing in the number of documents, we believe this is acceptable, because we do not expect n to exceed two hundred documents.

We consider it rather unhelpful to present the user with a deep classification structure. First, during our preliminary experiments we found large classification trees very distracting. A "tall" classification tree contains a large number of classes with just a few objects inside. Not only does the user have a hard time understanding the content of small classes, but also attention is diverted to the interpretation of the relationships between classes and subclasses. Second, the classification algorithm is not perfect and makes mistakes. The deeper we go in the clustering hierarchy, the more similar documents we encounter. The more similar the documents are, the more difficult it is for the algorithm to accurately cluster them. This increases the chances for the algorithm to misclassify the documents that, in its turn, results in even more confusing subtree configurations. Finally, a large structure simply encumbers the screen space, make it more difficult for the user to grasp the overall result. So, we limited every algorithm to produce clusters without any subclass hierarchy.

It is also unnecessary to classify every one of the retrieved documents. We observed that for a database with such diverse content as WallStreet87, approximately from one quarter to one third of the retrieved documents represent a unique subtopic. An effort to group these documents forces creation of

very strange and ambiguous classes and distorts the classification tree. In our experiments we allowed for significantly different documents to remain unclustered.

3.2.1 Single-link

Both van Rijsbergen and Willett [2, 16] named the single-link method as the one producing good results for the document clustering. The algorithm brings together documents which pairwise similarity exceeds a predefined threshold. For an object to belong to a cluster it needs to be similar enough to at least one other member of the cluster. To characterize the similarity between a pair of documents we used the cosine coefficient.

3.2.2 CLASSIT/AGGLOM

The most well-known conceptual clustering system is COBWEB [11, 17]. It creates clusters that are characterized by the list of nominal attribute values and probabilities associated with them. COBWEB's evaluation function, *category utility* [11, 14], estimates not the similarity between individual objects, but the overall quality of the partition.

In our experiments, we use the successor to COBWEB (CLASSIT [17]), that extends these ideas onto continuous attribute values. We had to modify CLASSIT's evaluation function to account for the missing terms [13]:

$$CU(\{C_1, C_2, \ldots, C_K\}) = \frac{\sum_{k}^{K} P(C_k) \frac{1}{I_k} \sum_{i}^{I_k} P(A_i|C_k) \, 1/\sigma_{ik} - \frac{1}{I} \sum_{i}^{I} P(A_i) \, 1/\sigma_i}{K},$$

where $P(C_k)$ is the probability to observe an instance from the class C_k, $P(A_i|C_k)$ and $P(A_i)$ are the probabilities that an attribute will be observed in the class C_k and in the dataset, I_k and I are the numbers of attributes in the given class C_k and in the dataset, K is the number of classes in the partition, σ_{ik} and σ_i are the standard deviations of attribute values for a given attribute in a given class and in the dataset.

CLASSIT creates classification trees that are strongly depend upon the order in which the objects are presented to the system. Fisher and his colleagues [12] suggested AGGLOM, an agglomerative version of COBWEB, that does not have this deficiency. We combined this algorithm with modified category utility function and named the new procedure CLASSIT/AGGLOM.

3.2.3 InCLASS

In addition to CLASSIT/AGGLOM and single-link we considered a mixed approach. Its main difference from the single-link method is that for every cluster it creates a cluster description by summing up profiles of its members. Like AGGLOM, the algorithm begins by creating a singleton cluster for every

document. It then proceeds by finding and merging the most similar clusters. Instead of evaluating the particular partitions, the algorithm estimates inter-cluster similarity using the cosine coefficient. A predefined threshold limits the documents that are too dissimilar from being clustered.

3.3 Cluster Descriptions

To describe a cluster we select a number of the most important terms from its members and present them to the user. The "important" terms may be defined in several possible ways. First, we could simply select the most frequent terms (the terms with the highest df) from the cluster. The advantage of this method is its independence from the clustering algorithm. Another approach is to rely on the evaluation function to rank the terms.

For example, consider the CLASSIT / AGGLOM system. A cluster can be defined as a partition of its members. The category utility characterizes the overall quality of the partition and, therefore, describes the quality of the cluster itself. We rewrote the category utility as

$$CU\left(\{C_1, C_2, \ldots, C_K\}\right) = \frac{1}{IK}\sum_{i}^{I} CT_i, \text{ where}$$

$$CT_i = \sum_{k}^{K} P(C_k)\frac{I}{I_k} P(A_i|C_k)\,1/\sigma_{ik} - P(A_i)\,1/\sigma_i \,,$$

The terms could be ranked according to their CT_i values, characterizing their input to the overall cluster quality.

The alternative is to replace the important terms with important phrases. A phrase is defined as sequence of one or more nouns. The underlining idea is that phrases are generally carry more content information than individual terms. To select the phrases from the documents we used InFINDER [18]. We discarded every phrase that appeared in more than fifty percent of the documents in the retrieved set.

4 Evaluation

The question is how one can evaluate a clustering system? The most obvious solution is to compare the output of the system, the automatically created classes, with a given standard. For our experiments, we created several sets of hand-built clusters. We ran the INQUERY system on the WallStreet87 database. During each run a set of fifty documents was retrieved. One of the authors read through these documents and divided them into classes to the best of his knowledge and understanding. It might not be easy to define a 'good' cluster even from a human point of view, therefore, several possible groupings were written down. Every automatically created cluster was compared against these hand-build classes to find the best match. We sum the number of matched (N_c) and unmatched (N_w) documents over the

whole set of clusters. The difference between the two sums ($N_c - N_w$) gives us the evaluation figure for the classification structure.

The main problem with this method is in obtaining the "gold" standard: it proved to be a rather expensive process. An alternative approach is based on the assumption, which is closely related to the Cluster Hypothesis [2]: relevant documents tend to be more similar to each other than to nonrelevant documents. In other words, relevant documents share a topic that is different from the topics of nonrelevant documents. Thus, clustering algorithm should put relevant and nonrelevant documents into separate classes. We define *separation factor* (S) as

$$S = \frac{\sum_{k=1}^{K} \max\left(Rel_k, Non_k\right)}{\sum_{k=1}^{K} \left(Rel_k + Non_k\right)},$$

where K is the number of clusters, Rel_k and Non_k are the number of relevant and nonrelevant documents in k^{th} cluster.

One may also envision the retrieved documents as surrounding the query, where the relevant documents are close to the center, than nonrelevant ones. Then, we may assume that the relevant documents are more similar to each other than nonrelevant and, therefore, we should see more relevant documents clustered. We redefine the well-known IR metrics recall (R) and precision (P) as the proportion of relevant documents that is clustered and as the proportion of clustered documents that are relevant.

$$R = \frac{\sum_{k=1}^{K} Rel_k}{Rel}, \; P = \frac{\sum_{k=1}^{K} Rel_k}{\sum_{k=1}^{K} \left(Rel_k + Non_k\right)},$$

where Rel is the number of relevant documents in the retrieved set.

The advantage of this approach is its minimal cost. We could apply this method using existing test collections, queries, and relevance judgments [10].

The last alternative is to perform usability testing on the system. This was not done for this paper.

4.1 Efficiency

INQUERY does not store document representatives. Therefore, the current version of the browser performs all computations during run-time. This process includes three main steps: reading and parsing the documents to extract the lists of terms; creating document profiles; and clustering the profiles. We conduct some efficiency testing by measuring the time required by the system for different phases of the clustering process.

5 Results and Discussion

5.1 Document Representation

We ran the InCLASS algorithm varying values for N_{use} from 25 to 150, $N_{discard}$ from 0 to 125, and threshold from 0.1 to 0.5. Table 1 shows several best scores averaged over the eight test sets.

N_{use}	$N_{discard}$	threshold	N_c	N_w	$N_c - N_w$
50	75	0.20	25.2	4.5	20.8
75	75	0.20	25.2	4.9	20.4
50	50	0.20	25.1	4.9	20.2
75	50	0.20	25.5	5.4	20.1
125	75	0.25	23.4	3.9	19.5
75	100	0.20	23.8	4.5	19.2
125	50	0.25	23.2	4.1	19.1
50	25	0.25	23.2	4.2	19.0
100	50	0.25	22.8	4.1	18.6
150	75	0.25	23.1	4.6	18.5

Table 1. Shows the results of running InCLASS on the eight test sets for various values of N_{use}, $N_{discard}$, and threshold.

We may conclude that it is enough to keep 50-100 of the best terms from a document for clustering purposes. Moreover, increasing the length of the profiles may degrade system performance.

Table 2 shows the best results obtained by running the InCLASS algorithm with three different term weighting schemes. Again, the scores are averaged over the eight document sets. For comparison we selected the best possible score for any clustering parameters on each test set and averaged these figures over the eight document sets. These are the numbers in brackets.

term weighting scheme	N_c	N_w	$N_c - N_w$
tf	21.8 (25.0)	4.4 (2.8)	17.4 (22.2)
*tf * idf*	14.6 (19.1)	2.0 (3.6)	12.6 (15.5)
*tf * dv*	20.6 (22.4)	5.6 (2.8)	15.0 (19.6)

Table 2. Shows the results of running InCLASS with different weighting schemes.

We assessed the importance of title terms by gradually increasing their weight values. Definite improvement was observed due to higher weights of the titles. However, there is also a possible danger in overestimating this value. After some point, the title terms become overweighted and the performance starts to decline.

We compared these results with the extreme case when only the title is used to represent the document and the text body is ignored altogether. The last row in the Table 3 clearly shows that the title alone is far from enough to define the document. The term frequency weighting scheme was used in these experiments.

weighting scheme	N_c	N_w	$N_c - N_w$
text + title	21.8 (25.0)	4.4 (2.8)	17.4 (22.2)
text + title * 3	23.0 (25.5)	3.9 (3.0)	19.1 (22.5)
text + title * 5	25.2 (26.5)	4.5 (3.0)	20.8 (23.5)
text + title * 10	20.1 (24.8)	2.9 (4.1)	17.2 (20.6)
tittle alone	12.0 (12.0)	9.8 (5.5)	2.2 (6.5)

Table 3. Shows the results of running InCLASS with different title weighting schemes.

5.2 Algorithms

All three algorithms were run on the eight data sets. Table 4 presents the results of these experiments. Here we used term frequency weighting scheme and weight values for the title terms were increased by the factor of 5.

algorithm	N_c	N_w	$N_c - N_w$
single-link	22.0 (25.2)	4.2 (4.1)	17.8 (21.1)
CLASSIT/AGGLOM	15.2 (29.2)	13.6 (18.1)	1.6 (11.1)
InCLASS	25.2 (26.5)	4.5 (3.0)	20.8 (23.5)

Table 4. Shows the results of running three different algorithms.

Here the InCLASS shows some improvements over the traditional single-link method. To our surprise, CLASSIT/AGGLOM was not even close to the former two. We believe it happens due to several reasons. First, CLASSIT was designed and tested for the objects without any missing attributes. A document profile has approximately 5% of nonzero entries. We believe, even with the category utility modifications the algorithm is still having a hard time handling the very sparse vectors. Second, this procedure is oriented toward maximizing predictive accuracy. Therefore, the concept tree it constructs may not reflect the structure underlying the training set.

We also compared the algorithms using the modified recall and precision metrics. For this comparison we choose five different techniques: single-link, CLASSIT/AGGLOM, InCLASS, InCLASS with title terms reweighted by factor of 5, and InCLASS using only titles. First, every algorithm was run on the eight test document sets and the best set of clustering parameters was selected. Then, the algorithms were run on the retrieved results produced by fifty TREC [10] queries on the WallStreet87. Finally, the metrics were computed using the relevance information, that accompanied the queries. Table 5 shows the means and standard deviations for S, R, and P averaged over fifty queries. These results are also summarized on Fig. 2.

algorithm	separation factor	modified recall	modified precision
single-link, title * 5	0.84±0.07	0.57±0.18	0.36±0.19
CLASSIT/AGGLOM, title * 5	0.84±0.08	0.56±0.26	0.32±0.20
InCLASS, title * 5	0.86±0.05	0.64±0.18	0.35±0.20
InCLASS, title * 1	0.85±0.06	0.54±0.18	0.35±0.20
InCLASS, title only	0.86±0.18	0.21±0.17	0.38±0.28

Table 5. Shows the results of running three different algorithms.

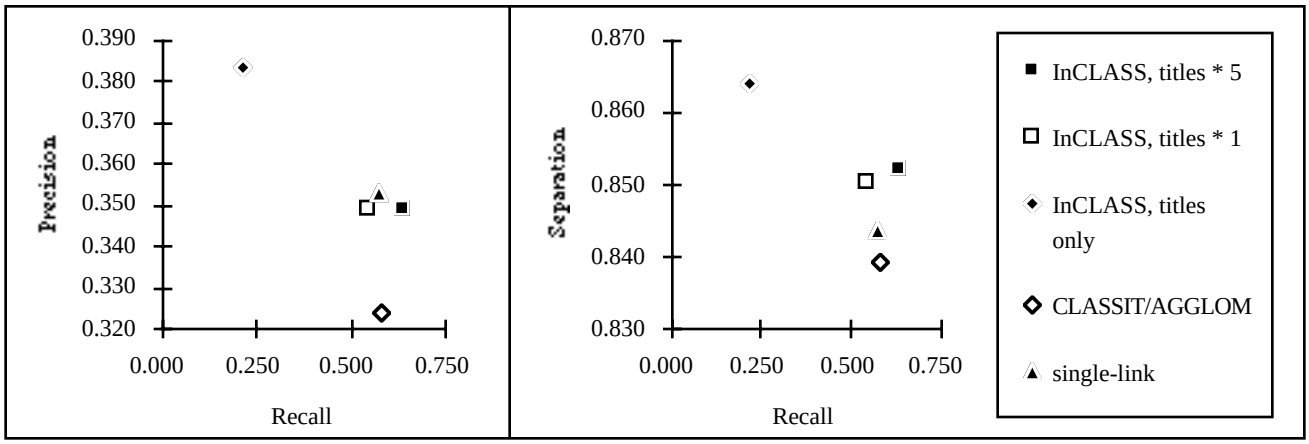

Fig. 2. Precision versus Recall and Separation versus Recall for five different techniques.

InCLASS with only titles was slightly better in separating relevant from nonrelevant documents, but it is well behind in recall. It does cluster significantly less documents than all other algorithms. InCLASS with reweighted titles was the best in recall. Although, this is not conclusive proof of the superiority of the InCLASS method, the overall results correspond to the ones obtained with hand-built classes.

5.3 Cluster Descriptions

Fig. 3. shows an example of a cluster created by the CLASSIT/AGGLOM algorithm. We provide the list of documents forming this cluster and the cluster descriptions as the list of ten terms weighted using *df* (bold-underlined typeface) and CT_i (italic-underlined typeface) factors. It seems rather difficult to prefer one technique over the other. One might select the *df* technique because of its generality. The most important feature both methods share is that they both create rather vague and difficult to read descriptions of the class content. We believe this happens due to several reasons. First, every such description consists exclusively of terms, and the individual terms do not usually carry much of the content value. Second, these terms are the stemmed words that were truncated without regard to the actual word meaning. Moreover, the stemming process does not always create correct word roots, forcing the reader to stumble over the words like *israe, influenc, offens,* etc.

drive, offens, border, area, day, defens, slow, claim, line, place

offens, f, gerald, border, drive, troop, basra, seib, area, defens

2. Confident Iran Boasts of War Advances As Iraq Mounts Ever-Bloodier Reprisals --- By Gerald F. Seib...
3. Iran's Drive on Basra Slows Despite Its Fighters' Esprit --- By Gerald F. Seib Staff Reporter of The Wall...
8. Iraq Is Said to Demolish Kurd Villages --- Diplomats Say Regime Seeks To Dent Rebels' Sway --- By...
38. Tehran Days: The Visitor to Iran Sees Very Little of the War, Much of Normal Life --- The Iran-Iraq...

leader, contra, deep, ayatollah, currenc, august, averag, boost, contract, daili

leader, israel, contra, deep, moslem, currenc, chang, rate, daili, shiite

19. How Iran Replenishes Its Deep War Chest --- By Dilip Hiro
29. Iran Aims to Be Superpower of the Gulf --- By Gerald F. Seib
30. Arab and Israeli Officials Tell Shultz U.S. Should Stage Counterattack on Iran --- By John Walcott Staff...
39. Washington Wire: A Special Weekly Report From The Wall Street Journal's Capital Bureau --- Compiled...

intellig, basra, neighbor, predict, aid, battl, final, troop, victor, day

elliot, karen, leader, deput, doubt, prime, basra, intellig, victor, offens

4. Iraq Says U.S. Disinformation About Iran Cost Thousands of Soldiers' Lives in Prolonged War --- By ...
5. Iraqi Minister Predicts a Basra Victory Will Limit Iranian Offensives for 1987 --- By Karen Elliott House...
7. Iran's Assault on Basra Poses Critical Test for Region --- Neighboring Arab States Fear Victory Could...
13. Title missing

quota, output, opec, level, cartel, deleg, benchmark, ceil, fail, produc

opec, quota, output, jame, tanner, petroleum, jr, petzing, thoma, cartel

16. OPEC Intends To Form Accord By Weekend --- Cartel Expects Iran to Bow Quickly or Not at All On Oil...
17. OPEC May Give Iraq a Quota as High As Iran's in Effort to Control Oil Output --- By James Tanner Staff...
23. OPEC Panel Seeks to Freeze Oil Price at $18 --- Market Committee Report Is Expected to Endorse...
40. OPEC Fails To Win Iran, Iraq Support --- Remaining 11 Oil Ministers Are Ready to Set Pact Without Gulf...
42. OPEC Accord Is Near but Faces Threats by Iran --- Cartel Would Slash Output, Hold $18-a-Barrel Price;...

reflag, flag, depart, didn't, question, congress, bill, relat, thing, terror

flag, reflag, water, kemp, frederick, thought, uss, congress, didn't, put

6. U.S. Appeals To Iraq to Show Restraint in War --- Officials, Gulf Allies Fear Results of New Attacks On...
14. Washington Insight: Iran Exploits U.S. Reflagging in Gulf To Build Morale, Counter Iraqi Threat --- By...
15. Washington Insight: Departing Iraqi Ambassador to U.S. Mastered Art of Wooing Americans --- By...
25. How to End the Gulf's Tanker War --- By Frederick Kempe
27. REVIEW & OUTLOOK (Editorial): Flags and Purposes
28. The U.S. and War in the Persian Gulf... --- By James A. Bill
34. Letters to the Editor: The Gulf War Isn't Flagging

Fig. 3. Shows an example of CLASSIT/AGGLOM output on the results from *Iran and Iraq* query.

Fig. 4 gives an example of InCLASS output. Here we used document frequency weighting scheme to select the best ten terms from the clusters. These descriptions are printed in the bold-underlined typeface. The second approach was to select ten the most frequent phrases using InFINDER (Jing, 1994). We discarded every phrase that appeared in more than fifty percent of the documents. The reader may see the final descriptions in the italic-underlined typeface. It seems that the phrases work better than the single terms. The third description, printed in bold-italic-underlined typeface was created by removing every phrase that appears as a part of another phrase. That made the descriptions slightly more detailed and eliminated repetitions like *iraqis, iraqi, iraqi troop,* etc.

One may still stumble on the irregularly stemmed phrases like *oil reserv*. This is the result of using the Porter stemmer [19] in the current version of the system. This stemmer employs several heuristics for truncating the words. It does not take into account the word meaning and usually produces irregular and sometimes cryptic stems. Fig. 5 presents an example of InCLASS output produced on the same WallStreet87 database build with KSTEM [20]. This stemmer was specially designed to overcome the deficiencies of its predecessor. It is impossible to accurately compare the cluster descriptions, because

both indexing and query matching was done with the new stemmer and the retrieval results and classes are different from what we have seen before. However, one may clearly notice that the descriptions in this example consist of real words. We are currently working on a system that uses both KSTEM and phrases for class representations.

basra, offens, hussein, diplomat, leader, troop, victor, soldier, fear, saddam

basra, iranians, staff reporter, baghdad, iranian troop, iranian forc, iraqis, iraqi, iranian citi, usa arms sale

basra, staff reporter, baghdad, iranian troop, iranian forc, iranian citi, usa arms sale, iraqi troop, iraqi offic, iraqi territor

1. Baghdad's Goal: Iraq's Aim in Gulf War Is No Longer to Win But to Avoid Losing -- Hussein Employs His…
2. Confident Iran Boasts of War Advances As Iraq Mounts Ever-Bloodier Reprisals -- By Gerald F. Seib…
3. Iran's Drive on Basra Slows Despite Its Fighters' Esprit -- By Gerald F. Seib Staff Reporter of The Wall…
5. Iraqi Minister Predicts a Basra Victory Will Limit Iranian Offensives for 1987 -- By Karen Elliott House …
7. Iran's Assault on Basra Poses Critical Test for Region -- Neighboring Arab States Fear Victory Could …

flag, mine, editor, target, navi, avoid, stark, kemp, pentagon, strike

washington, usa offic, kuwaiti tanker, iranian attack, iranian, iran-iraq war, baghdad, iraqi attack, iranians, ussr

washington, usa offic, kuwaiti tanker, iranian attack, iran-iraq war, baghdad, iraqi attack, ussr, uss stark, soviets

6. U.S. Appeals To Iraq to Show Restraint in War -- Officials, Gulf Allies Fear Results of New Attacks On…
9. Iraq Renews Attacks on Iran Targets In Gulf, Creating Perils for U.S. Navy -- By Tim Carrington Staff…
10. *Title missing*
11. Letters to the Editor: Iran-Iraq War Could Engulf Everyone
14. Washington Insight: Iran Exploits U.S. Reflagging in Gulf To Build Morale, Counter Iraqi Threat -- By …
24. REVIEW & OUTLOOK (Editorial): The Stark Attack
25. How to End the Gulf's Tanker War -- By Frederick Kempe
27. REVIEW & OUTLOOK (Editorial): Flags and Purposes
34. Letters to the Editor: The Gulf War Isn't Flagging
37. U.S. Is Tilting Toward Iraqis In the Gulf War -- Aide May Go to Baghdad As White House Tries To Regain…
43. Pentagon Spells Out Persian Gulf Plans But Policy Continues to Come Under Fire -- By Tim Carrington…
47. *Title missing*
49. REVIEW & OUTLOOK (Editorial): Roiling the Gulf

opec, quota, output, produc, accord, ibrahim, 18, youssef, sourc, jame

opec, organization, oil price, oil minist, production quota, petroleum exporting countries, staff reporter, king fahd, youssef m ibrahim, opec member

organization, oil price, oil minist, production quota, petroleum exporting countries, staff reporter, king fahd, youssef m ibrahim, opec member, oil export

16. OPEC Intends To Form Accord By Weekend -- Cartel Expects Iran to Bow Quickly or Not at All On Oil…
17. OPEC May Give Iraq a Quota as High As Iran's in Effort to Control Oil Output -- By James Tanner Staff…
21. OPEC's Push For $18 Oil Price Remains Stalled -- Continued Refusal by Iraq To Join Pact on Output …
23. OPEC Panel Seeks to Freeze Oil Price at $18 -- Market Committee Report Is Expected to Endorse …
32. OPEC Calls Meetings on Overproduction Amid New Signs That Pact Is Weakening -- By James Tanner…
33. Iraq Rejects Saudi King's Plea on Plan To Cut OPEC Output, Threatening Pact -- By Youssef M. Ibrahim…
36. Twelve Members of OPEC Agree to Cut Oil Output, But Iraq Resists the Accord -- By Youssef M.…
40. OPEC Fails To Win Iran, Iraq Support -- Remaining 11 Oil Ministers Are Ready to Set Pact Without Gulf…
41. Group Aims to Reestablish Dominance by Cutting Oil Output, Fixing Price -- By Youssef M. Ibrahim…
42. OPEC Accord Is Near but Faces Threats by Iran -- Cartel Would Slash Output, Hold $18-a-Barrel Price; Mediators Begin Work -- By Thomas Petzinger Jr. and James Tanner Staff Reporters of The Wall…
45. OPEC Is Approaching Midyear Parley -- Oil Prices Firm, but Iraq Keeps Tough Stance -- By Youssef M. …
48. Iraq Is Posing Threat to OPEC Oil-Price Policy -- Arab Nation's Plan to Boost Its Capacity for Exports …
50. OPEC Accord to Cut Daily Oil Output By One Million Barrels Appears Close -- By Youssef M. Ibrahim …

Fig. 4. Shows an example of InCLASS output on the results from *Iran and Iraq* query.

israel, lebanon, change, relations, ayatollah, moslem, public, islam, victory, moderate

21. Washington Insight: Departing Iraqi Ambassador to U.S. Mastered Art of Wooing Americans --- By…
24. Iran Aims to Be Superpower of the Gulf --- By Gerald F. Seib
26. Arab and Israeli Officials Tell Shultz U.S. Should Stage Counterattack on Iran --- By John Walcott …
34. Title missing
49. U.S. Gain From an Iranian Victory --- By Michael Reisman

navy, mine, flag, carrington, tim, pentagon, strike, renew, hit, aircraft

8. Iraq Renews Attacks on Iran Targets In Gulf, Creating Perils for U.S. Navy --- By Tim Carrington …
11. Title missing
17. Title missing
18. Dire Straits: U.S. Ponders Response To Any Iran Reprisals For Shielding Tankers --- As Tehran's …
31. Pentagon Spells Out Persian Gulf Plans But Policy Continues to Come Under Fire --- By Tim …
45. U.S. Navy Jet Launched Two Missiles At Iranian Warplane Judged 'Hostile' --- By Tim Carrington …

barrel, production, quota, opec, 18, meeting, james, tanner, output, agreement

14. OPEC Intends To Form Accord By Weekend --- Cartel Expects Iran to Bow Quickly or Not at All On …
19. Title missing
23. OPEC May Give Iraq a Quota as High As Iran's in Effort to Control Oil Output --- By James Tanner …
33. OPEC Panel Seeks to Freeze Oil Price at $18 --- Market Committee Report Is Expected to Endorse…
37. Twelve Members of OPEC Agree to Cut Oil Output, But Iraq Resists the Accord --- By Youssef M. Ibrahim Staff Reporter of The Wall Street Journal
38. OPEC Calls Meetings on Overproduction Amid New Signs That Pact Is Weakening --- By James …
39. OPEC's Push For $18 Oil Price Remains Stalled --- Continued Refusal by Iraq To Join Pact on Output …
43. OPEC Fails To Win Iran, Iraq Support --- Remaining 11 Oil Ministers Are Ready to Set Pact Without …
44. OPEC Accord Is Near but Faces Threats by Iran --- Cartel Would Slash Output, Hold $18-a-Barrel Price;…

Fig. 5. Shows an example of InCLASS output on the results from *Iran and Iraq* query using KSTEM.

5.4 Efficiency

The following table shows how much time each one of the clustering phases takes on a PowerMac 7100/80. All experiments were done on the WallStreet87 database. The numbers are averaged over the eight test document sets.

The time required by the system to read and parse the documents exceeds the actual clustering time by a factor of 100. It usually takes more than a minute to cluster a hundred documents. INQUERY allows the document texts to be cached during the retrieval phase. The memory space requirement rises to allow all documents be stored, but the speed increases dramatically as well. We observed five times speed improvement with caching turned on.

Number of documents	Time (in sec.)			
	Read and parse		Create profiles	Cluster
	caching off	caching on		
50	34.0	6.1	0.1	0.3
100	71.2	11.9	0.2	1.1

Table 6. Shows the time required by different clustering steps on a PowerMac 7100/80.

We estimate that it might be possible to get even more improvement by creating and storing the document vectors during the database indexing phase. In this case there will be no time spent on the reading and parsing phase, and the whole clustering process will take just about a second. The disadvantage of this approach is that storage overhead requirements are increased. For example,

WallStreet87 contains approximately 44,000 documents. Allowing a hundred terms per profile will give us around 33.5 Mb increase in storage space. This is close to a quarter of the original database size (128 Mb).

6 Summary

Our experiments show, that

- Keeping 50-100 of the most frequent terms is sufficient for document representation. Moreover, increasing the proportion of terms included in the profiles seems to degrade the system performance. This observation allows us to reduce the requirements on computational resources. It also means that all calculation could be done independently from the document size.
- Using term frequency weighting in the document representation works better, or at least as well as the more sophisticated techniques. One may prefer this scheme due to its simplicity and minimal computational cost.
- Taking into account the high contextual value of titles improves the system performance. This can easily achieved by increasing the weights of the title terms. However, there is a possible danger in overestimating this value.
- Agglomerative, hierarchical $O(n^2)$ clustering technique are robust, sufficiently fast and produce reasonable classification structures.
- Phrasal cluster representations are superior compared to individual terms. The quality of the stemmer plays important role in the generation of cluster descriptions.
- The suggested evaluation methods produce reasonable estimations of the classification quality and generally agree with an expert's opinion. One should prefer the comparison with hand-build classes against the other techniques as more human oriented. However, this method is very expensive and the other techniques could be used for preliminary study.

The general conclusion is that the suggested techniques show some promise and work reasonably well. However, our experiments were done on a relatively small domain. It remains to be seen how these methods behave on different collections and with different document sizes. A more accurate evaluation requires more user experimentation.

7 Future Work and User Interface Ideas

The immediate advantage the classes could give the user is to facilitate the relevance feedback process. Defining a cluster as relevant will augment the query and direct the search process toward the class topic. Technically it would mean adding a relevance checkbox to the cluster descriptor. Marking the cluster as relevant is equivalent to marking as relevant every document in the cluster.

The clustering algorithm has a set of parameters that defines the form of the document profiles, limits the values of the similarity function, and, therefore, determines the clustering hierarchy. It is expected that the user will generally rely on the preset values of these parameters. We consider giving the user more immediate control over the cluster parameters. That could be done in a form of sliders or scrollbars located in the vicinity of the clusters. Aside from giving the user the ability to fine-tune the system, these controls would serve as the means for exploring the similarity among the documents and the clusters. Sliding the control and observing the system regrouping the documents, the user would improve his/her understanding of the cluster contents.

The current version uses background coloring to designate the class membership. The colors are randomly selected from a given palette. We observed that it definitely improves the user's perception of the clusters and we are considering extending and systematizing this approach. The task would be to define a base palette with a set of colors to designate clusters. Then the relations inside the cluster might be explored by changing the color's intensity according to similarity between the document and the class descriptor. For example, the more different the document from the class descriptor the lighter its background becomes. Also a similar technique may be adopted to present the similarities among documents from the different classes. In this case the document color becomes a mixture of basic cluster colors in the proportion of the document similarity to every one of them.

Finally, we are considering giving the user some means to correct, or even completely change the classification structure. Setting the cluster boundaries, for example, by direct manipulation of the document icons on the screen, the user would be able to define his own classification. Then, the algorithm is to learn from this experience, e.g., by adjusting the term weights to reflect the new information. For instance, the algorithm brought together two documents. They both have a topic in common, but each of them also has a secondary theme. Breaking apart this cluster the user would shift the system attention to these secondary topics, probably affecting the other classes as well.

References

1. Hanson, S. and Bauer, M.. Conceptual clustering, categorization, and polymorphy. *Machine Learning*, 3, 343-372; 1989.

2. van Rijsbergen, C. J.. Information Retrieval. London: Butterworths; 1979.

3. Salton, G.. Automatic Text Processing. Addison-Wesley; 1989.

4. Cutting, D. R., Karger, D. R. and Pederson, J. O. Constant interaction-time Scatter/Gather browsing of very large document collections. In *Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 126-134; 1993.

5. Rao, R., Pederson, J. O., Hearst, M. A., and others. Rich interaction in the digital library. *Communications of the ACM*. Vol. 38, No. 4. pp. 29-39; 1995.

6. Michalski, R. S. Knowledge acquisition through conceptual clustering: a theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems, 4,* 219-243; 1980.

7. Gowda, K. C. and Krishna, G. Disaggregative clustering using the concept of mutual nearest neighborhood. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-8, No. 12, pp. 888-894; 1978.

8. Callan, J. P. Use of domain knowledge in constructive induction. Technical Report 90-95, Computer Science Department, University of Massachusetts at Amherst; 1990.

9. Callan, J. P. Passage-level evidence in document retrieval. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* pp. 302-310. Dublin, Ireland; 1994.

10. Harman, D. The TIPSTER evaluation corpus. CDROM disks of computer readable text. 1992. Available from the Linguistic Data Consortium.

11. Fisher, D. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2, 139-172; 1987.

12. Fisher, D., Xu, L., Zard, N. Ordering effects in clustering. In *Proceedings of the Ninth International Machine Learning Conference.* 163-168; 1992.

13. Weinberg, J. B., Biswas, G., Koller, G. R. Conceptual clustering with systematic missing values. In *Proceedings of the Ninth International Machine Learning Conference.* 464-469; 1992.

14. Gluck, M. A. and Corter, J. E. Information, uncertainty, and the utility of categories. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 283-287). Irvine, CA: Lawrence Erlbaum Associates; 1985.

15. Salton, G. *The SMART Retrieval System.* Englewood Cliffs, NJ: Prentice-Hall; 1971.

16. Willett, P. Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management.* Vol. 24, No. 5, pp. 577-597; 1988.

17. Gennari, J. H., Langley, P., and Fisher, D. Models of incremental concept formation. *Artificial Intelligence*, 40, pp. 11-61; 1989.

18. Jing, Y. and Croft, W. B. An association thesaurus for information retrieval. In *RIAO 94 Conference Proceedings*. pp. 146-160. New York; 1994.

19. Porter, M. F. An algorithm for suffix stripping. *Program*. Vol. 14, pp. 130-137; 1980.

20. Krovetz, R. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* pp. 191-202; 1993.

21. Cutting, D. R., Pederson, J. O., Karger, D. R. and Tukey, J. W. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* pp. 318-329; 1992.